US005404446A

United States Patent [19]

Bowater et al.

[11] Patent Number: 5,404,446

[45] Date of Patent: Apr. 4, 1995

[54] DUAL BUFFER VIDEO DISPLAY SYSTEM FOR THE DISPLAY OF ASYNCHRONOUS IRREGULAR FRAME RATE VIDEO DATA

[75] Inventors: Ronald J. Bowater, Romsey; Barry K. Aldred, Winchester; Steven P. Woodman, Romsey, all of England

[73] Assignee: International Business Machines Corporation, Armonk, N.Y.

[21] Appl. No.: 37,197

[22] Filed: Mar. 26, 1993

[30] Foreign Application Priority Data

Mar. 26, 1992 [GB] United Kingdom 9206554

[51] Int. Cl.[6] G06F 15/20

[52] U.S. Cl. 395/162

[58] Field of Search 395/101, 162, 164, 200, 395/250; 358/85, 86, 183, 903; 370/62; 380/18; 382/56; 345/185, 186, 189, 192, 196, 200

[56] References Cited

U.S. PATENT DOCUMENTS

4,027,100	5/1977	Ishiguro	178/69.1
4,394,774	7/1983	Widergren et al.	382/56
4,949,169	8/1990	Lumelsky et al.	358/86
4,995,071	2/1991	Weber et al.	379/53
5,014,267	5/1991	Tompkins et al.	370/62
5,062,136	10/1991	Gattis et al.	380/18
5,192,999	3/1993	Graczyk et al.	358/85
5,262,965	11/1993	Putnam et al.	395/101

FOREIGN PATENT DOCUMENTS

0031031	7/1981	European Pat. Off.	G06F 13/00
0382182	8/1990	European Pat. Off.	G06F 13/12
0274703	7/1988	Germany	G06F 3/06
8600482	1/1986	WIPO	H04L 25/49

OTHER PUBLICATIONS

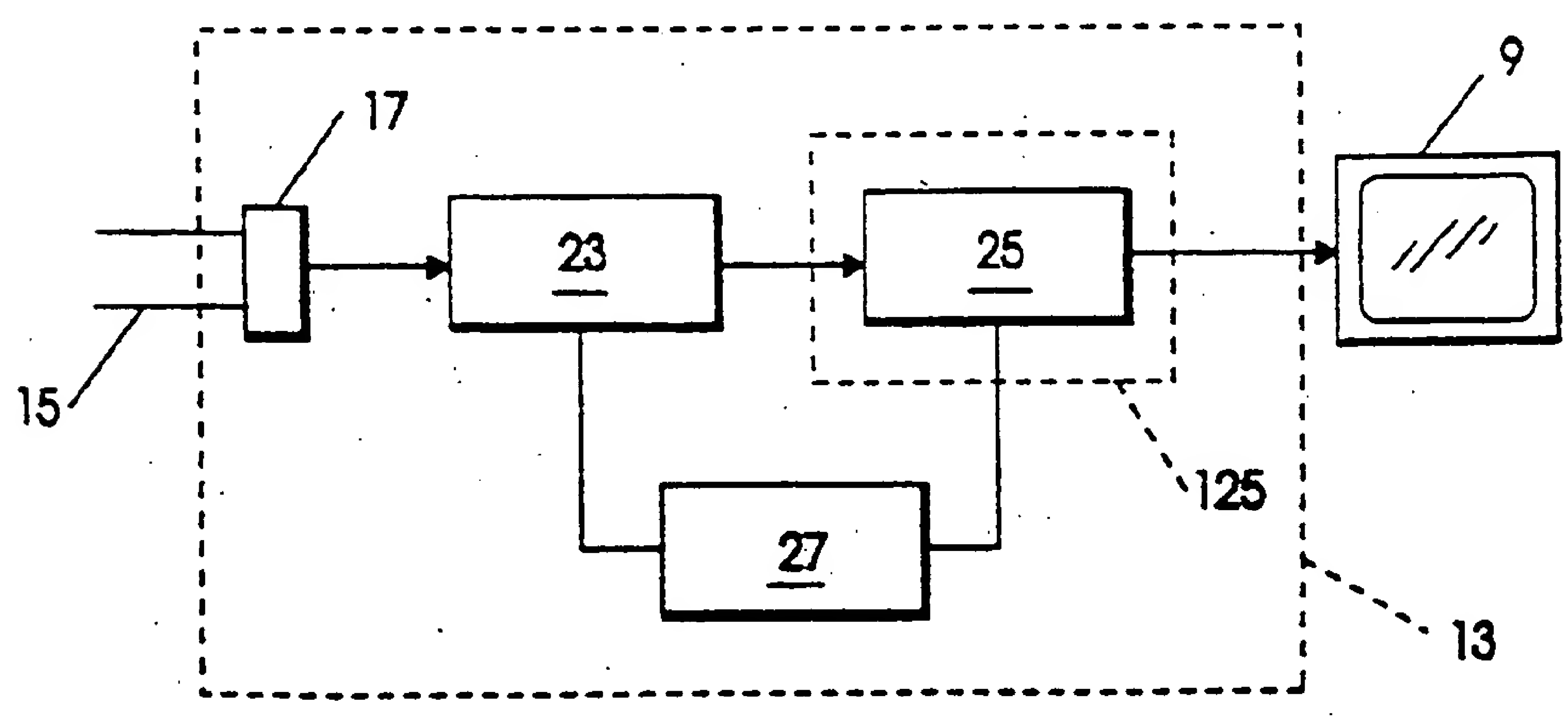
SMPTE Journal, vol. 85, Jun. 1976, pp. 385–388, Matley, "A Digital Framestore Synchronizer".

IBM Technical Disclosure Bulletin, vol. 10, No. 1, Jun. 1967, pp. 34–36, H. E. Jenkins et al., "Asynchronous Control of Data Transfer".

Primary Examiner—Mark R. Powell
Assistant Examiner—U. Chauhan
Attorney, Agent, or Firm—Martin J. McKinley

[57] ABSTRACT

7 Claims, 1 Drawing Sheet



Docket no. 2134

EXHIBIT A

FIG. 1

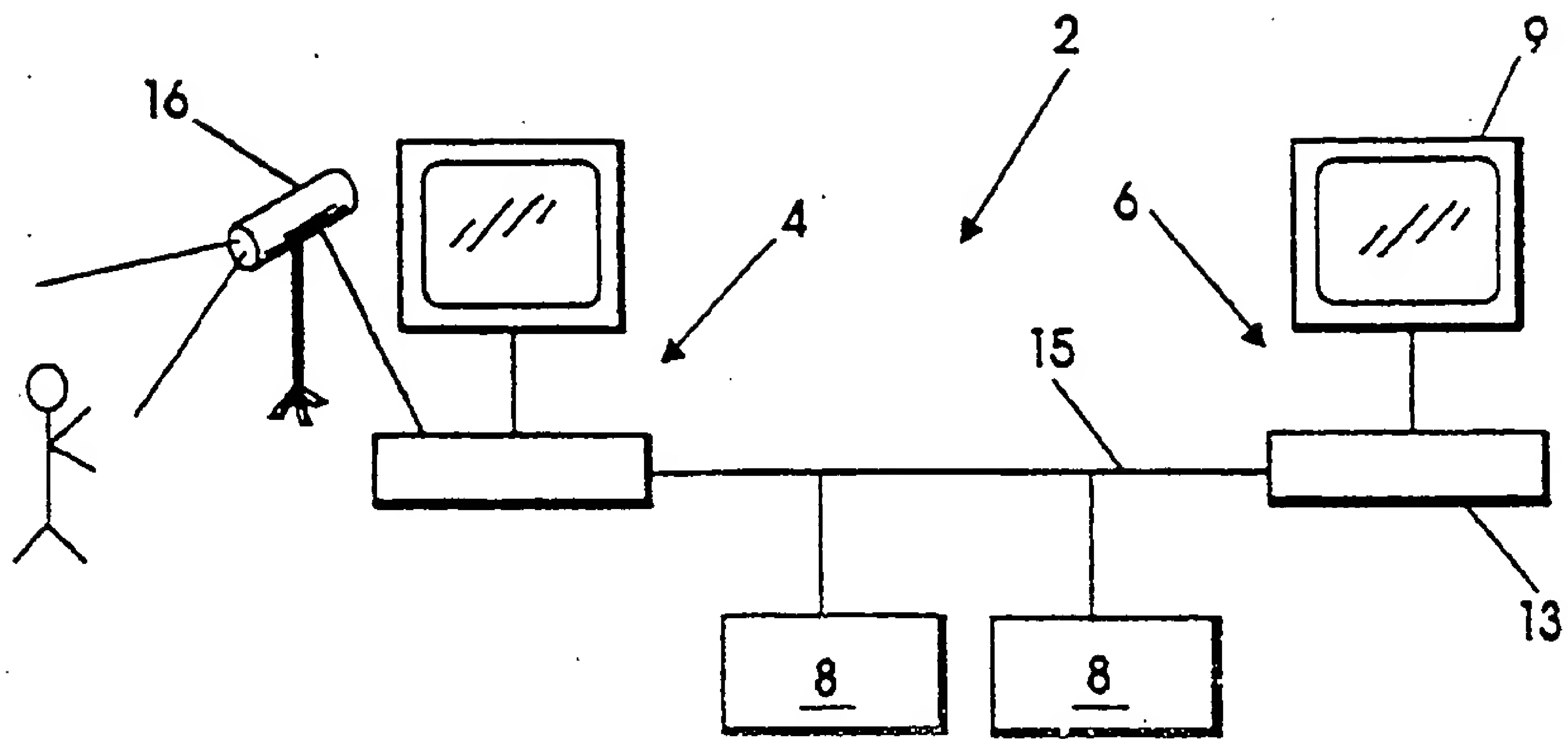
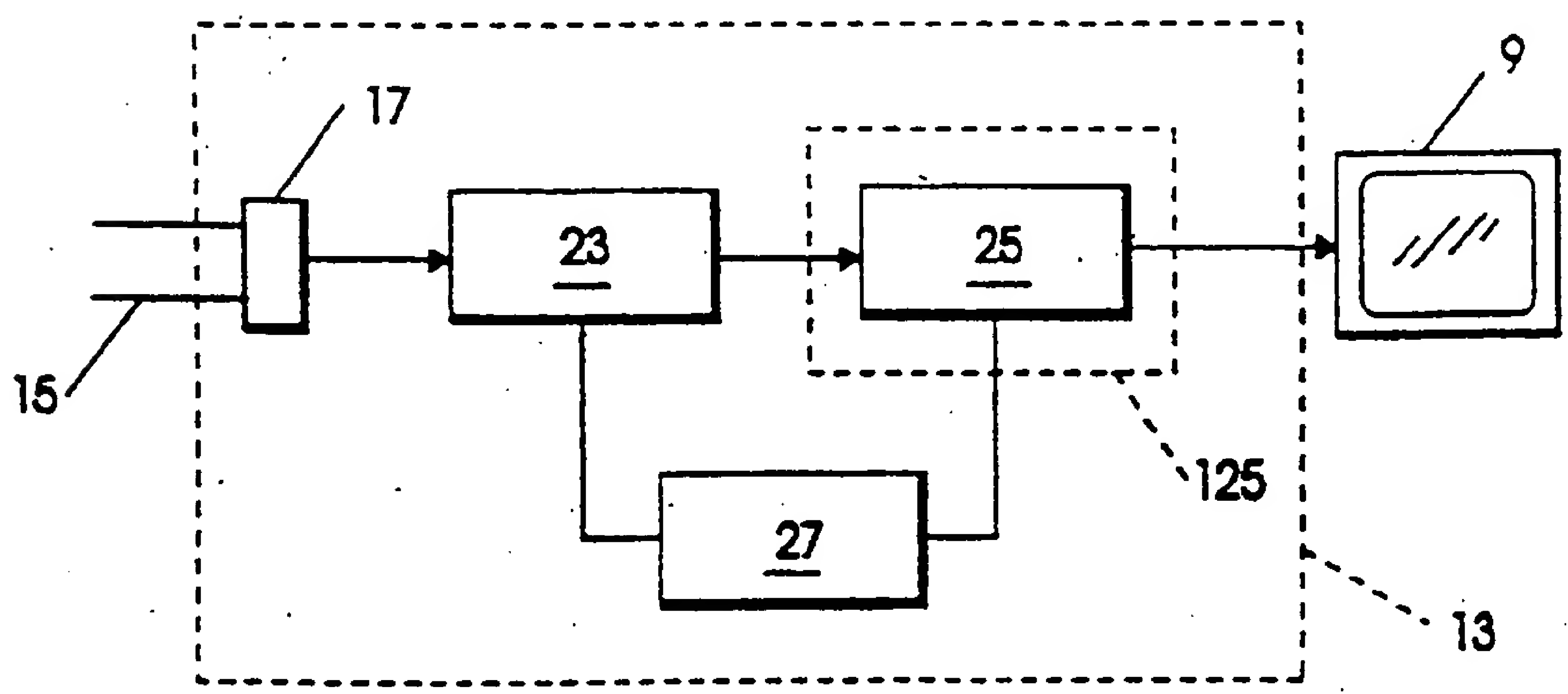


FIG. 2



Docket no. 2134 **EXHIBIT A**

DUAL BUFFER VIDEO DISPLAY SYSTEM FOR THE DISPLAY OF ASYNCHRONOUS IRREGULAR FRAME RATE VIDEO DATA

BACKGROUND OF THE INVENTION

In computer-based video communication systems, a video signal is obtained from the camera at a constant frame rate but, after transmission across the asynchronous or non-ideal network, the frames arrive at irregular intervals. Some frames arrive early, some are delayed, and bunching can occur. The display device at the receiving terminal, however, generally requires a constant frame rate supplied to it (e.g., to match the raster scan rate of a CRT). In such systems it is therefore necessary to match the irregular arrival of frames over the network with the constant supply required to the output screen.

. The designer of computer based video communication systems is therefore faced with the problem of how to achieve regular play-out of the asynchronous incoming video signal while, at the same time, minimising the number of buffered video frames.

Docket no. 2134

EXHIBIT A

FIG. 1 shows a network 2 of computers 4, 6, 8 connected by an asynchronous communication channel 15 (e.g., LAN or ISDN). A camera 16 at a first computer 4 obtains a video signal, normally of the user, which is compressed and submitted to the network. The signal is then transmitted down the communication channel in packet format before arriving at the destination computer 6. Typically, this second computer includes hardware such as the Intel/IBM ActionMedia II (AMII) card, which is responsible for actually decompressing and displaying the video image on the screen 9. In video conferencing applications, the reverse process also occurs; i.e., the second computer is simultaneously sending an image of its user back to the first computer 4 for display. It is also possible to set up multi-way conferences.

With reference now to FIG. 2, the incoming video signal from the communication subsystem 15 arrives at the workstation 13 for display on the associated monitor 9. The signal is transferred first to a buffer 23, and then to the AMII card 125 or, more particularly, to the AudioVisual Kernel (AVK) interface buffer 25 of the AMII card. The buffer 23 provides a FIFO queue, conveniently implemented as a circular buffer. A control process 27 is responsible first for reading incoming data into the circular buffer, and then for transferring data from the circular buffer to the AVK.

Video images are captured at the source computer at a frame rate of 15 frames per second (in this particular embodiment), which is sufficient to provide moderate quality video. This is also the rate at which they are read out of the AVK to the screen. However, the transmission rate over the network is variable, depending on network load, etc., so that the arrival rate at the end of the computer subsystem departs from this 15 Hz clock. Changes in CPU activity at the source and destination computers can also lead to variations in the effective frame arrival rate. Individual frames can have either a positive or negative offset from their nominal arrival time, although it is assumed that frames do, in fact, arrive in the correct sequence. It should be noted that the variation in arrival times is such that, even if the hardware could display each frame directly on arrival, the resulting sequence would be so temporally distorted as to be unwatchable.

essential.

Together, the AVK and circular buffer compensate for the variable arrival rate of the video frames by introducing a time-lag, T(L), between the received and displayed images. Any frame arriving within T(L) of its nominal arrival time can be properly displayed. Only if a frame arrives more than T(L) late, will the AVK and circular buffer empty and the video image will freeze. To decrease the risk of buffer starvation, the buffer size can be increased to make T(L) larger, but with a 15 frames per second transmission rate, storing only 10 frames adds a delay of $\frac{2}{3}$ second. If the effectiveness of interactive applications such as video conferencing is not to be seriously degraded, only a handful of frames can be buffered with T(L) correspondingly small.

The control process is responsible first for receiving data into the circular buffer, and then for forwarding it to the AVK. There is no control over output from the AVK, which is at a fixed rate. As explained in more detail below, the AVK requests frames from the circular buffer as required. Clearly, if frames are present in the circular buffer, then these can be forwarded to the AVK. However, in video conferencing or other interactive applications where the overall amount of buffering is limited, there may occasionally be particularly long delays on the network during which time the circular buffer empties. In this case, the control process reacts by loading the AVK with null frames. A null frame is essentially the same as the preceding frame, so that, as far as the viewer is concerned, video image temporarily freezes. Thus, each time the control process fails to find frames in the circular buffer, the requisite number of null frames are loaded into the AVK instead.

Although the user may not notice the insertion of individual null frames, each null frame adds to the overall delay in the system (i.e., it is effectively another form of buffering). If more and more null frames are inserted into the video stream, then this will, again, lead to an intrusive delay between transmission and display. This problem can be overcome by the circular buffer throwing away real data when the delayed frames do finally arrive. These frames are then effectively lost, allowing the displayed image to catch up with the incoming signal. It is the presence of two buffers that gives the flexibility to lose frames in this way, and so cope with occasional delays longer than T(L).

The technique used to discard frames exploits the fact that, due to the limited bandwidth of the channel, the video signal is compressed before transmission over a computer based communication line. Basically, two types of compression, spatial and temporal, are used. In the former, the redundancy within a single frame is removed, for example, by using the fact that adjacent pixels often have closely related brightness and color values. A frame encoded using only spatial compression is known as a "still frame". Temporal compression achieves a further level of compression by exploiting the fact that the luminosity and color of the same pixel in two consecutive frames are, again, likely to be highly correlated. Therefore, in temporal compression, a frame is encoded as a "relative frame" in terms of its difference from the previous frame (we assume that a relative frame is also spatially compressed). The greatest reduction in data is achieved if every frame (apart from the first) is a relative frame, but this is highly error prone since the loss of a single frame will produce defects that persist for all subsequent frames. Therefore, as a compromise, every Nth frame can be sent as a still frame, with all intervening frames as relative frames, so that the result of compression is a regularly spaced series of frames whose size varies somewhat according to the temporal and spatial content of the data and, of course, whether that particular frame is a still or relative frame.

When the buffer is not full, then incoming frames can be added to the buffer in the normal way. However, when the buffer is full, there are two possible actions. If the incoming frame is a still frame, then the entire buffer is flushed before the incoming still frame is added to the queue. Alternatively, if the incoming frame is a relative frame, then only relative frames below (i.e., that arrived earlier than) a still frame are flushed. This is because the previous still frame is still required to make sense of the relative frames. In either case, flushing the buffer results in some frames being thrown away, and so the displayed image catches-up slightly with the received image.

Turning now to the AVK, frames are read out from the AVK for display at a fixed rate. This leads to the possibility of buffer starvation if the AVK contains no more frames to read out to the screen. In such an eventuality, the AVK pipeline needs to be reset, requiring a considerable system overhead during which time the video image is not updated, in contrast to the circular buffer, which can be emptied and refilled without penalty.

Accordingly, a lower limit, V(L), is set for the number of frames in the AVK. This value is selected to substantially preclude buffer starvation yet, at the same time, not introduce an unacceptable delay. The control process responsible for transferring frames from the circular buffer to the AVK then tries to maintain the number of frames in the AVK as close as possible to but slightly above V(L).

Once the control process has determined the number of frames to transfer to the AVK, it can either send this as a single request, or as an appropriate number of requests for individual frames. In the latter case, the circular buffer can respond simply to each request by transferring a frame if available, or inserting a null frame if not.

Docket no. 2134

EXHIBIT A

* * * * *